

Documento de propuesta de desarrollo de
software I, II y III

DUNGEON SCHOOL

Emilio José Plaza Valdés eplazavaldes@correo.unicordoba.edu.co
Maria Lucia Dilso Mosquera mdilsomosquera@correo.unicordoba.edu.co
Brayan Andres Ramos Fuentes bramosfuentes42@correo.unicordoba.edu.co
Diego Andrés Salas Álvarez dasalasalvarez@correo.unicordoba.edu.co

Tutor: Alexander Toscano Ricardo



Descripción del videojuego educativo

Se busca crear y diseñar un videojuego original para ordenadores ya sea flash o instalable que permita la evaluación y diseño de estas en diferentes áreas centradas en temas específicos.

Dichas evaluaciones dependiendo el área estarán divididas en 2 sesiones donde en la primera se centrarán en preguntas esparcidas por un mapa que parodia a un salón de clases donde también abran ítems de ayuda de manera aleatoria y la segunda sección se centrará en la representación de un maestro como jefe final donde estará el grueso de la evaluación que se está aplicando de igual manera se podrán usar ítems sobrantes de primera sección en la parte del jefe.

El videojuego tendrá temas y preguntas prefabricadas y de diferentes dificultades de igual manera se buscará en próximas actualizaciones que estas puedan ser más personalizables al igual que los temas centrales o que se implemente la capacidad de integrada por IA de generar preguntas pertinentes en base a un plan de periodo o plan de clases.

ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS	6
1. INTRODUCCIÓN	6
PROPÓSITO DEL DOCUMENTO	6
ALCANCE DEL PROYECTO MÓDULO DE PIZARRA COMPARTIDA	10
DEFINICIONES Y ACRÓNIMOS.....	12
2. DESCRIPCIÓN GENERAL	13
OBJETIVOS DEL SISTEMA.....	13
FUNCIONALIDAD GENERAL.....	13
USUARIOS DEL SISTEMA	14
RESTRICCIONES.....	15
3. REQUISITOS FUNCIONALES	15
CASOS DE USO	15
DIAGRAMAS DE FLUJO DE CASOS DE USO	17
DESCRIPCIÓN DETALLADA DE CADA CASO DE USO	17
PRIORIDAD DE REQUERIMIENTOS	18
4. REQUISITOS NO FUNCIONALES	19
REQUISITOS DE DESEMPEÑO	19
REQUISITOS DE SEGURIDAD.....	19
REQUISITOS DE USABILIDAD	19
REQUISITOS DE ESCALABILIDAD.....	19
5. MODELADO E/R.....	20
DIAGRAMA DE ENTIDAD-RELACIÓN	20
DIAGRAMA RELACIONAL.....	20
SCRIPT DE MODELO RELACIONAL	21
DESCRIPCIÓN DE ENTIDADES Y RELACIONES.....	21
REGLAS DE INTEGRIDAD REFERENCIAL	21
COLECCIONES (NOSLQ).....	22
6. ANEXOS	23
DIAGRAMAS ADICIONALES	23
REFERENCIAS	23
ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND	24
7. INTRODUCCIÓN	24
PROPÓSITO DE LA ETAPA	24
ALCANCE DE LA ETAPA	24
DEFINICIONES Y ACRÓNIMOS.....	24
8. DISEÑO DE LA ARQUITECTURA DE BACKEND	24
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA.....	24
COMPONENTES DEL BACKEND	24

DIAGRAMAS DE ARQUITECTURA.....	24
9. ELECCIÓN DE LA BASE DE DATOS	24
EVALUACIÓN DE OPCIONES (SQL o NoSQL)	25
JUSTIFICACIÓN DE LA ELECCIÓN.....	25
DISEÑO DE ESQUEMA DE BASE DE DATOS.....	25
10. IMPLEMENTACIÓN DEL BACKEND	25
ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN	25
CREACIÓN DE LA LÓGICA DE NEGOCIO.....	25
DESARROLLO DE ENDPOINTS Y APIS	25
AUTENTICACIÓN Y AUTORIZACIÓN.....	25
11. CONEXIÓN A LA BASE DE DATOS.....	25
CONFIGURACIÓN DE LA CONEXIÓN	26
DESARROLLO DE OPERACIONES CRUD.....	26
MANEJO DE TRANSACCIONES	26
12. PRUEBAS DEL BACKEND	26
DISEÑO DE CASOS DE PRUEBA	26
EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN	26
MANEJO DE ERRORES Y EXCEPCIONES	26
ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND	27
13. INTRODUCCIÓN.....	27
PROPÓSITO DE LA ETAPA	27
ALCANCE DE LA ETAPA	27
DEFINICIONES Y ACRÓNIMOS.....	27
14. CREACIÓN DE LA INTERFAZ DE USUARIO (UI)	27
DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS.....	27
CONSIDERACIONES DE USABILIDAD.....	27
MAQUETACIÓN RESPONSIVA.....	27
15. PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS)	27
DESARROLLO DE LA LÓGICA DEL FRONTEND.....	28
MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS	28
USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE)	28
16. CONSUMO DE DATOS DESDE EL BACKEND	28
CONFIGURACIÓN DE CONEXIONES AL BACKEND.....	28
OBTENCIÓN Y PRESENTACIÓN DE DATOS	28
ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE)	28
17. INTERACCIÓN USUARIO-INTERFAZ	28
MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS.....	29

IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS.....	29
MEJORAS EN LA EXPERIENCIA DEL USUARIO	29
18. PRUEBAS Y DEPURACIÓN DEL FRONTEND	29
DISEÑO DE CASOS DE PRUEBA DE FRONTEND	29
PRUEBAS DE USABILIDAD.....	29
DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO.....	29
19. IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND.....	29
MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO)	30
VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND.....	30
20. INTEGRACIÓN CON EL BACKEND	30
VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND.....	30
PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND	30
ANEXOS	30

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

1. Introducción

Propósito del Documento

El presente documento tiene como finalidad documentar el proceso de diseño, análisis e implementación del videojuego educativo en su etapa inicial. El desarrollo se basará en la metodología SECMALI, la cual permitirá estructurar y validar cada fase del proyecto para garantizar su efectividad pedagógica y técnica.

El documento se divide en tres etapas para facilitar su entendimiento y aplicación en la asignatura de diseño de software.

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Diseño de la Aplicación y Análisis de Requisitos, Esta fase se enfoca en definir los elementos fundamentales del videojuego educativo, considerando su estructura, mecánicas y requerimientos técnicos. El objetivo es establecer una base sólida que permita desarrollar una experiencia lúdica centrada en el aprendizaje y la evaluación de conocimientos en un entorno simulado tipo aula escolar.

El videojuego estará dividido en dos secciones principales:

1. Exploración del mapa:

El jugador recorrerá un entorno en pixel art que representa un aula de clases. A lo largo del mapa encontrará preguntas educativas de dificultad variada, así como ítems de ayuda dispersos aleatoriamente que podrán ser usados más adelante.

2. Enfrentamiento con el maestro (jefe final):

Tras completar la exploración, el jugador se enfrentará al “jefe final”, representado por un maestro. En esta fase se evaluarán los conocimientos adquiridos mediante preguntas más complejas, y se podrán utilizar los ítems recolectados previamente para facilitar el progreso.

Objetivos del diseño:

- Evaluar el aprendizaje de los estudiantes de forma dinámica y entretenida.
- Fomentar el uso de la tecnología como recurso pedagógico.
- Motivar a los usuarios mediante retos progresivos y retroalimentación visual.
- Crear un entorno inmersivo adaptado a la estética pixel art.

Requerimientos funcionales:

- Sistema de preguntas por niveles o temas.
- Registro de respuestas correctas e incorrectas.
- Uso estratégico de ítems recolectados.
- Interfaz intuitiva para navegación y combate.
- Escenarios interactivos con diseño en pixel art.

Requerimientos no funcionales:

- Compatibilidad con ordenadores de bajos recursos.
- Bajo consumo de ancho de banda en caso de conexión al servidor.
- Facilidad de actualización y personalización de preguntas.
- Diseño atractivo y accesible para estudiantes de distintos niveles.

Etapas 2: Persistencia de Datos con Backend – Servidor

En esta etapa se desarrollará el backend que permitirá almacenar y gestionar la información generada durante el uso del videojuego. Esto incluye el progreso de los jugadores, respuestas seleccionadas, estadísticas de juego y configuraciones personalizadas.

Objetivos:

- Crear una API que permita el almacenamiento y recuperación de datos del juego.
- Implementar una base de datos para guardar el progreso del jugador, resultados por nivel, preguntas y configuración de ítems.
- Facilitar la personalización futura del contenido por parte de docentes (subida de preguntas propias, configuraciones de dificultad).

Componentes:

- Endpoints para el registro y login de usuarios.
- Registro de progreso por sesión.
- Almacenamiento de preguntas y resultados por jugador.
- Carga dinámica de datos (preguntas, niveles, ítems).

Tecnologías sugeridas:

- Lenguaje: JavaScript o TypeScript.
- Framework: Express.js o Nest.js.
- Base de datos: MongoDB (NoSQL) o PostgreSQL (SQL), según necesidades.
- API: RESTful con autenticación mediante JWT.
- Herramientas adicionales: Mongoose (si se usa MongoDB), Prisma (si se usa PostgreSQL).

Etapas 3: Consumo de Datos y Desarrollo Frontend – Cliente

Esta etapa contempla la implementación de la interfaz visual y funcional del videojuego en estilo pixel art, centrado en brindar una experiencia lúdica y educativa atractiva.

Objetivos:

- Desarrollar una interfaz gráfica en pixel art que represente un aula y escenarios de batalla con el jefe final.
- Implementar mecánicas de exploración, interacción con preguntas y uso de ítems.
- Conectar el cliente con el backend para cargar preguntas, enviar respuestas y registrar resultados.

Características:

- Sección de exploración estilo RPG 2D con estética pixel art.
- Mecánicas para recolectar ítems y resolver preguntas en un entorno interactivo.
- Sección de jefe final donde se aplican preguntas más complejas.
- Uso de los ítems recolectados como ayudas o potenciadores.
- Feedback visual retro acorde a la estética pixel art (barras de energía, cuadros de texto tipo RPG).

Tecnologías sugeridas:

- Motor/Framework para 2D en pixel art: Phaser.js o Godot (GDScript) si se quiere mantener libre de Unity.
- Lenguaje: JavaScript (Phaser) o GDScript (Godot).
- Recursos visuales: Sprites 2D en pixel art, tilesets para el mapa del aula, animaciones sencillas.
- Sonido: Efectos retro y música chiptune para ambientar.

Alcance del Proyecto Dungeon School

Dungeon School es una propuesta innovadora de videojuego educativo multiplataforma que busca revolucionar el aprendizaje en entornos digitales. Mezclando mecánicas de juegos de rol con elementos pedagógicos, este proyecto se plantea como una herramienta divertida e interactiva para reforzar conocimientos en el aula o desde casa. Ambientado en un universo escolar con estética pixel art retro, el juego permitirá a los estudiantes recorrer mapas tipo aula, resolver retos académicos, recolectar ítems útiles y enfrentarse a un "maestro jefe" para demostrar lo aprendido.

El objetivo principal es motivar a los estudiantes a aprender jugando, brindando una experiencia inmersiva y significativa. Además, el juego está diseñado para expandirse con el tiempo, integrando funciones cada vez más complejas que permitan adaptar el contenido a distintos niveles educativos, personalizar la experiencia del usuario y fomentar el trabajo colaborativo. Este documento detalla el alcance inicial y propone una clasificación de funcionalidades futuras según la complejidad que implicaría su desarrollo e implementación.

Funcionalidades actuales:

- Movimiento del personaje (Arriba, abajo, izquierda, derecha)
- Desbloqueo de zonas
- Interacción con preguntas
- Recolección de ítems
- Uso de ítems
- Enfrentamiento con un enemigo
- Registro de respuestas
- Estado del personaje (barras de energía, barra de estado, etc.)

Funcionalidades futuras:

Funciones Básicas (fácil implementación)

- Guardar el progreso del jugador automáticamente.
- Personalizar el avatar del estudiante con ropa o accesorios.
- Elegir entre distintos mapas escolares para explorar.
- Obtener pistas al ver contenido educativo dentro del juego.
- Subir puntuaciones a un ranking global o escolar.

- Conseguir medallas por logros educativos.
- Cambiar la dificultad del juego según el nivel del estudiante.
- Escuchar narraciones o audios explicativos como ayuda extra.
- Compartir resultados o certificados con padres o docentes.
- Añadir minijuegos entre preguntas para mantener la motivación.
- Personalizar el fondo o estilo del aula virtual.
- Acceder a estadísticas personales de rendimiento.
- Guardar partidas y continuar desde donde se dejó.

Funciones Medias (requieren planificación y desarrollo intermedio)

- Sistema de progresión con logros.
- Versión portable offline.
- Desbloquear niveles temáticos (ciencias, historia, etc.).
- Usar objetos especiales para resolver preguntas más rápido.
- Participar en torneos académicos entre colegios.
- Chatear con otros jugadores dentro del aula virtual.
- Usar animaciones o reacciones del maestro jefe según el rendimiento.
- Integrar videos educativos antes de ciertas preguntas clave.
- Permitir a los profes crear mini misiones con recompensas.
- Recibir retroalimentación automática según errores comunes.
- Jugar con teclado, mouse, pantalla táctil o control.
- Descargar preguntas desde una nube educativa segura.
- Acceder a una biblioteca con guías de estudio integradas.
- Usar un sistema de energía para regular el tiempo de juego.

- Hacer misiones diarias o semanales con objetivos específicos.

Funciones Complejas (altamente personalizadas o tecnológicamente avanzadas)

- Personalización de pruebas por parte del docente.
- Generación automática de preguntas con IA según el plan de clase.
- Modo multijugador (en tiempo real o por turnos).
- Participar en eventos mensuales con recompensas especiales.
- Usar comandos por voz para responder o navegar.
- Desbloquear zonas secretas si se alcanza cierto puntaje.
- Usar monedas virtuales para desbloquear contenido educativo extra.

Definiciones y Acrónimos

- API: Interfaz de Programación de Aplicaciones (Application Programming Interface).
- DBMS: Sistema de Gestión de Bases de Datos (Database Management System).
- JWT: Token Web JSON (JSON Web Token), utilizado para autenticar usuarios.
- CRUD: Crear, Leer, Actualizar y Borrar (Create, Read, Update, Delete).
- Pixel Art: Estilo gráfico basado en píxeles visibles, común en videojuegos retro.
- RPG: Juego de Rol (Role-Playing Game), estilo de juego donde el jugador asume el papel de un personaje en una narrativa.

2. Descripción General

Objetivos del Sistema

El objetivo del sistema es proporcionar una pizarra compartida dentro de un Sistema de Gestión de Contenido llamado CREAVI que permita a los usuarios colaborar de manera eficiente y efectiva, facilitando la creación, visualización y edición de contenido visual en tiempo real. Esta pizarra compartida se diseñará con el propósito de mejorar la comunicación y la colaboración en un entorno en línea, ofreciendo a los usuarios una plataforma intuitiva y versátil para crear y compartir ideas, diagramas, esquemas y contenido visual de manera colaborativa, enriqueciendo así la experiencia de usuario y la productividad en el uso del CMS.

Funcionalidad General

- **Creación y Edición Colaborativa:** Permite a los usuarios crear y editar contenido en la pizarra de forma colaborativa en tiempo real. Múltiples usuarios pueden trabajar en el mismo documento simultáneamente.
- **Herramientas de Dibujo y Anotación:** Proporciona herramientas de dibujo, pinceles, formas y opciones de anotación que permiten a los usuarios plasmar sus ideas y conceptos de manera visual.
- **Carga de Imágenes y Multimedia:** Permite a los usuarios cargar imágenes, videos y otros medios directamente en la pizarra, lo que facilita la ilustración de conceptos.
- **Organización de Contenido:** Ofrece opciones para organizar y estructurar el contenido en la pizarra, como la creación de capas, agrupación de elementos y uso de etiquetas.
- **Historial de Revisiones:** Registra un historial de revisiones que permite a los usuarios rastrear los cambios realizados en la pizarra y restaurar versiones anteriores si es necesario.
- **Compartir y Colaborar:** Permite compartir la pizarra con otros usuarios a través de enlaces o invitaciones, lo que facilita la colaboración con colegas, clientes o amigos.
- **Comentarios y Chat en Tiempo Real:** Los usuarios pueden comentar y discutir sobre el contenido de la pizarra a través de un chat en tiempo real, lo que facilita la comunicación durante la colaboración.
- **Exportación e Impresión:** Ofrece la capacidad de exportar el contenido de la pizarra en varios formatos (PDF, imagen, etc.) y la opción de imprimirlo.

- **Integración con el CMS:** Se integra de manera transparente con el sistema de gestión de contenido (CMS CREAVI), lo que permite incrustar pizarras en los contenidos, metodologías o cualquier otro tipo de componente que permita la pizarra.
- **Personalización y Temas:** Permite a los usuarios personalizar la apariencia de la pizarra y seleccionar temas que se adapten a sus necesidades.
- **Acceso Seguro:** Proporciona medidas de seguridad para garantizar que solo los usuarios autorizados puedan acceder y editar la pizarra.
- **Notificaciones y Actualizaciones en Tiempo Real:** Los usuarios reciben notificaciones sobre cambios en la pizarra y pueden ver actualizaciones en tiempo real mientras otros editan.
- **Acceso Móvil:** Ofrece una experiencia de usuario optimizada en dispositivos móviles, permitiendo el acceso y la colaboración desde smartphones y tabletas.
- **Búsqueda y Filtros:** Facilita la búsqueda de contenido en la pizarra y la aplicación de filtros para organizar y encontrar información específica.
- **Gestión de Usuarios y Permisos:** Permite a los administradores gestionar usuarios y definir permisos de acceso y edición.
- **Informes y Analíticas:** Proporciona información sobre el uso de la pizarra, como quién la ha editado, cuándo se realizaron cambios y estadísticas sobre el contenido(XAPI).

Usuarios del Sistema

Los siguientes usuarios pueden interactuar con la pizarra dependiendo de las funcionalidades.

Funcionalidad	Administradores	Docente	Alumno
Movimiento del personaje (Arriba, abajo, izquierda, derecha)	✓		✓
Desbloqueo de zonas	✓		✓
Interacción con preguntas	✓		✓
Recolección de ítems	✓		✓
Uso de ítems	✓		✓
Enfrentamiento con un enemigo	✓		✓
Registro de respuestas	✓	✓	✓
Estado del personaje (barras de energía, barra de estado, etc.)	✓		✓
Guardar el progreso	✓		✓
Personalizar el avatar	✓		✓
Unirse a una partida	✓	✓	✓

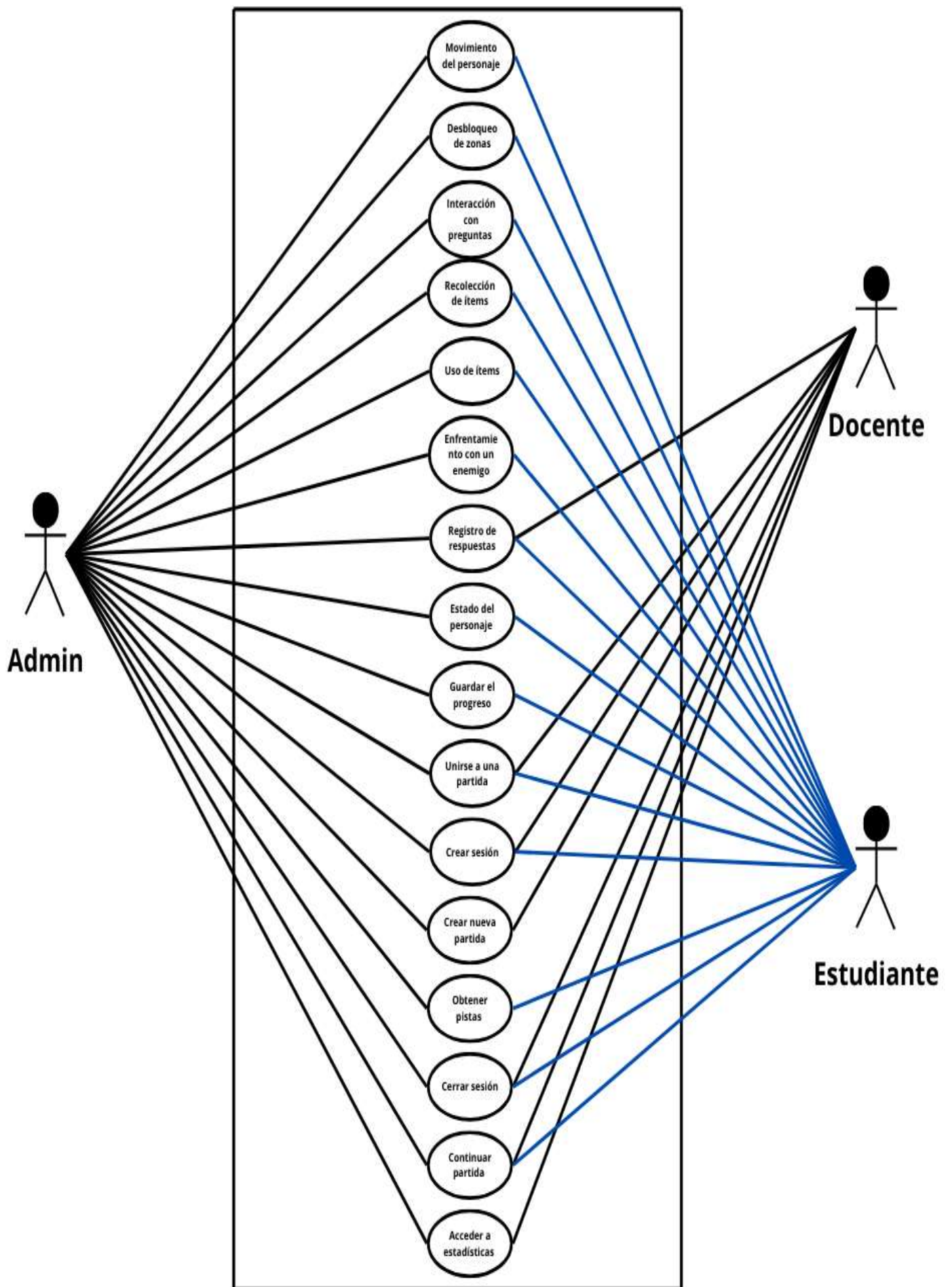
Crear nueva partida	✓	✓	
Elegir entre distintos mapas	✓	✓	✓
Obtener pistas	✓		✓
Ranking	✓	✓	✓
Conseguir medallas	✓		✓
Cambiar la dificultad	✓	✓	
Escuchar narraciones	✓		✓
Compartir resultados	✓	✓	✓
Continuar partida	✓	✓	✓
Personalizar el fondo	✓	✓	
Acceder a estadísticas	✓	✓	
Guardar partidas	✓	✓	✓
Crear sesión	✓	✓	✓
Cerrar sesión	✓	✓	✓

Restricciones

3. Requisitos Funcionales

Casos de Uso

Diagrama de caso de uso



Diagramas de Flujo de Casos de Uso

Descripción detallada de cada caso de uso

Prioridad de Requerimientos

4. Requisitos No Funcionales

Requisitos de Desempeño

Requisitos de Seguridad

Requisitos de Usabilidad

Requisitos de Escalabilidad

5. Modelado E/R

Diagrama de Entidad-Relación

Diagrama Relacional

Script de modelo relacional

<https://dbdiagram.io/>

Descripción de Entidades y Relaciones

Entidades:

Relaciones:

Reglas de Integridad Referencial

Colecciones (NoSLQ)

6. Anexos

Diagramas Adicionales

Referencias

Etapa 2: Persistencia de Datos con Backend

7. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

8. Diseño de la Arquitectura de Backend

Descripción de la Arquitectura Propuesta

Componentes del Backend

Diagramas de Arquitectura

9. Elección de la Base de Datos

Evaluación de Opciones (SQL o NoSQL)

Justificación de la Elección

Diseño de Esquema de Base de Datos

10. Implementación del Backend

Elección del Lenguaje de Programación

Creación de la Lógica de Negocio

Desarrollo de Endpoints y APIs

Autenticación y Autorización

11. Conexión a la Base de Datos

Configuración de la Conexión

Desarrollo de Operaciones CRUD

Manejo de Transacciones

12. Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

Etapa 3: Consumo de Datos y Desarrollo Frontend

13. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

14. Creación de la Interfaz de Usuario (UI)

Diseño de la Interfaz de Usuario (UI) con HTML y CSS

Consideraciones de Usabilidad

Maquetación Responsiva

15. Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

16. Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

17. Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

18. Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

19. Implementación de la Lógica de Negocio en el Frontend

Migración de la Lógica de Negocio desde el Backend (si necesario)

Validación de Datos y Reglas de Negocio en el Frontend

20. Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend

ANEXOS

Diagramas UML